AF/2182
RFW

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appellants:	Clive M. Philbrick et al.	Ser. No:	09/802,551
Filing Date:	March 9, 2001	Examiner:	A. Perez-Daple
Atty. Docket No:	ALA-012	GAU:	2182
Assignee:	Alacritech, Inc.		

For: INTELLIGENT NETWORK STORAGE INTERFACE SYSTEM

August 24, 2005

Mail Stop Appeal Brief
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

BRIEF FOR APPELLANTS

This is an Appeal of the Final Rejection of claims 1-18, 21 and 22 dated February 25, 2005. A Notice of Appeal was received by the Patent Office on June 24, 2005.

Real Party In Interest

Assignee Alacritech, Inc. is the real party in interest.

Related Appeals and Interferences

Appellants know of no other appeals or interferences that will directly affect or be directly affected by or have a bearing on the Board's decision in this pending Appeal.

Status of Claims

The application was originally filed with 20 claims. In response to a Restriction Requirement mailed August 27, 2004, applicants canceled claims 19 and 20, and added new claims 21 and 22. Pending claims 1-18, 21 and 22 stand rejected and are the subject of this Appeal. Appendix A lists the claims that are the subject of this Appeal.

Status of Amendments

No amendment has been filed subsequent to the Final Rejection.

Summary of Claimed Subject Matter[1]

Claim 1 recites an apparatus for transferring information between a network (FIG. 1; 25, 28) and a storage device (FIG. 1: 66, 70), the apparatus comprising: a host computer (FIG. 1: 20) having a CPU (FIG. 1: 30) operating a file system (FIG. 1: 23, p. 9, l. 24 - p. 10, l. 21) and a host memory (FIG. 1: 33) connected to said CPU by a host bus (FIG. 1: 35), and an interface device (FIG. 1: 22) coupled to said host computer, to the network and to the storage device, said interface device including an interface memory (FIG. 1: 46) containing an interface file cache (FIG. 1: 80) adapted to store data that is communicated between the network and the storage device under control of said file system (p.5, ll. 16-20; p. 10, ll. 12-21), wherein said host computer is configured to designate a User Datagram Protocol socket (p. 31, ll. 19-27) that is accessible by said interface device (p. 32, ll. 9-26), and said interface device is configured to communicate said data between the network and the file cache (p. 11, ll. 5-11; p. 12, ll. 22-28; p. 13, ll. 25-29; p. 33, ll. 1-20) according to said User Datagram Protocol socket (p. 32, ll. 9-28; p. 33, ll. 1-20).

Independent claim 11 recites an apparatus for transferring information between a network (FIG. 15: 604, 644, 650, 652) and a peripheral device (FIG. 15: 666, 667), the apparatus comprising: a host computer (FIG. 15: 600, 602) having a processor (FIG. 1: 30) connected to a host memory (FIG. 1: 33) by a host memory bus (FIG. 1: 35), said host memory containing an application operable by the processor to designate a User Datagram Protocol socket (p. 31, ll. 19-27), and an interface device (FIG. 1: 22; FIG. 15: 606, 622) connected to said host computer and coupled between the network and the peripheral device, said interface device including an interface memory (FIG. 1: 46) adapted to store data corresponding to said User Datagram Protocol socket (p. 32, ll. 9-

[1] The following summary pursuant to 37 CFR §41.37(c)(1)(v) is a concise explanation of the independent claims and is to be read in light of the disclosure. For conciseness this summary does not list all of the places in the specification and figures that relate to those claims. This summary does not limit the claims (see MPEP §1206).

28) and a mechanism configured to associate said data with a User Datagram Protocol header (p. 30, l. 26 – p. 32, l. 30) corresponding to said User Datagram Protocol socket (p. 32, ll. 9-28) such that said data is communicated between the network and the peripheral device without encountering said host computer (p. 32, ll. 9 – p. 34, l. 2).

Independent claim 21 recites an apparatus for transferring information between a network (FIG. 1; 25, 28) and a storage device (FIG. 1: 66, 70), the apparatus comprising: a host computer (FIG. 1: 20) having a CPU (FIG. 1: 30) operating a file system (FIG. 1: 23, p. 9, l. 24 - p. 10, l. 21) and a host memory (FIG. 1: 33) connected to said CPU by a host bus (FIG. 1: 35), and an interface device (FIG. 1: 22) coupled to said host computer, to the network and to the storage device, said interface device including an interface memory (FIG. 1: 46) containing an interface file cache (FIG. 1: 80) adapted to store data that is communicated between the network and the storage device under control of said file system (p.5, ll. 16-20; p. 10, ll. 12-21), wherein said host computer is configured to designate a User Datagram Protocol socket (p. 31, ll. 19-27) that is accessible by said interface device (p. 32, ll. 9-26), and said interface device has means for communicating said data between the network and the file cache (p. 11, ll. 5-11; p. 12, ll. 22-28; p. 13, ll. 25-29; p. 33, ll. 1-20) according to said User Datagram Protocol socket (p. 32, ll. 9-28; p. 33, ll. 1-20).

Grounds of Rejection to be Reviewed on Appeal

(1) The rejection of claims 1, 3-18 and 21 under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 5,913,028 to Wang et al. ("Wang") in view of Stevens, TCP/IP Illustrated, Volume 1: The Protocols, New York, 1994 ("Stevens") and further in view of Schulzrinne et al., RFC 1889, http://www.ietf.org/rfc/rfc1889.txt, January 1996 ("Schulzrinne").

(2) The rejection of claims 2 and 22 under 35 U.S.C. §103(a) as being unpatentable over Wang in view of Stevens and Schulzrinne and further in view of "Applicant's admitted prior art (pg. 31, line 28 – pg. 32, line 8) (hereinafter AAPA)."

Argument

I. Regarding Grounds of Rejection (1), the Final Rejection states:

Claims 1, 3-18 and 21 are rejected under 35 U.S.C. 103(a) as being unpatentable over Wang et al. (US 5,913,028) (hereinafter Wang) in view of Stevens ("TCP/IP Illustrated, Volume 1: The protocols" New York, 1994.) and further in view of Schulzrinne et al. (RFC 1889, http://www.ietf.org/rfc/rfc1889.txt?number=1889, January 1996) (hereinafter Schulzrinne).

As for claims 1 and 21, Wang discloses an apparatus for transferring information between a network and a storage device, the apparatus comprising:

a host computer having a CPU, (CPU 24, Fig. 2) operating a file system (file system 27, direct file system 28, and peer I/O manager 26, Fig. 3) and a host memory (memory 32, Fig. 2) connected to said CPU by a host bus (Figs. 2 and 3), and

an interface device coupled to said host computer, to the network and to the storage device, said interface device including an interface memory containing an interface file cache (local memory 44, Fig. 3) adapted to store data that is communicated between the network and the storage device under control of said file system (Network I/O device, Fig. 3; col. 3, lines 31-52),

wherein said host computer is configured to designate *a socket* that is accessible by said interface device, and said interface device is configured to communicate said data between the network and the file cache according to said *socket* (Note, a socket is merely an endpoint of a connection, which is inherently required for communications on a network. See cited definition from techdictionary.com.; col. 4, line 38 – col. 5, line 5; Fig. 3).

Although obvious to one of ordinary skill in the art at the time of the invention, Wang does not explicitly teach that the socket may be a Uniform Datagram Protocol (UDP) socket. The Examiner notes that UDP is a well-known protocol for data transfer on networks, as shown by Stevens, chapter 11, for example. Schulzrinne further teaches the use of UDP as an underlying protocol for RTP in order to maintain the real-time characteristics of data such as audio and video (section 1, Introduction). As understood by one of ordinary skill in the art at the time of the invention, when UDP is used, UDP sockets are inherently required in order to receive the information packets over the network (see cited definition from techdictionary.com). It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the teachings of Wang by using UDP sockets in order to maintain the real-time characteristics of data such as audio and video, as taught by Schulzrinne above.

A. Claim 1

1) Wang does not teach or suggest what the Final Rejection alleges Wang teaches.

Wang does not teach or suggest an "interface file cache," as recited in claim 1, which is "under control of said file system." As described in the present application, on page 5, lines 16-18:

> A file cache is provided on the interface device for storing data that may bypass the host, with organization of data in the interface device file cache controlled by a file system on the host.

As further described in the present application, on page 9, line 24- page 10, line 9:

> The file system 23 is a high level software entity that contains general knowledge of the organization of information on storage units 66 and 70 and file caches 24 and 80, and provides algorithms that implement the properties and performance of the storage architecture. The file system 23 logically organizes information stored on the storage units 66 and 70, and respective file caches 24 and 80, as a hierarchical structure of files, although such a logical file may be physically located in disparate blocks on different disks of a storage unit 66 or 70. The file system 23 also manages the storage and retrieval of file data on storage units 66 and 70 and file caches 24 and 80. I/O driver 67 software operating on the host 20 under the file system interacts with controllers 64 and 72 for respective storage units 66 and 70 to manipulate blocks of data, i.e., read the blocks from or write the blocks to those storage units. Host file cache 24 and INIC file cache 80 provide storage space for data that is being read from or written to the storage units 66 and 70, with the data mapped by the file system 23 between the physical block format of the storage units 66 and 70 and the logical file format used for applications. Linear streams of bytes associated with a file and stored in host file cache 24 and INIC file cache 80 are termed file streams. Host file cache 24 and INIC file cache 80 each contain an index that lists the file streams held in that respective cache.

In contrast, a "FILE CACHE 4a" is shown on the front page of Wang (see also FIG. 5) indicating that Wang and others of ordinary skill in the art know what a file cache is. The "FILE CACHE 4a" is shown as part of the "planar board 20" of Wang, and is not part of the "network I/O device 40." In other words, Wang does not show an "interface file cache" as alleged by the Final Rejection.

Wang also shows a "DIRECT FILE SYSTEM 28" and a "TRADITIONAL FILE SYSTEM 27" in FIG. 5, but does not in that figure or anywhere else teach or describe an

"interface file cache," as recited in claim 1, which is "under control of said file system." Instead, responsive to Appellants' demonstrating that deficiency in the Office Action, the Final Rejection alleges that "peer I/O manager 26" is a part of the file system, in contrast to the text and figures of Wang.

For example, in contrast to the Final Rejection allegation that "peer I/O manager 26, Fig. 3" is "a file system," Wang states in column 11, lines 19-21:

> The Peer I/O Manager, like NCP, is a protocol layer that accepts data packets addressed to its own unique Socket ID in a file server. The client redirector extension is a client-based software protocol layer that addresses file requests to the Peer I/O Manager socket ID in the file server. (emphasis added)

According to this passage of Wang, the "Peer I/O Manager" of Wang is a network protocol layer, and is not "a file system" as alleged by the Final Rejection. Moreover, Wang teaches that the "Peer I/O manager" has "its own unique Socket ID," as opposed to being "a file system" as alleged by the Final Rejection. Applicants respectfully assert that "a file system" would not have "its own unique Socket ID," but would instead be able to manage data for various applications.

This differentiation between the "Peer I/O Manager" and the "file system" of Wang is further described in column 11, lines 26-36 of that reference, which states:

> Thus, Peer I/O manager services co-exist with traditional file server functions the NCP protocol facilitates file open/close/read/write operations between traditional Network Attached Clients and file servers. NCP facilitates file read/write requests through Planar-board Centric I/O data transfers, i.e. all data moves through the File Cache in Planar-board memory. Peer I/O Manager, however, facilitates file read/write requests through Peer I/O data transfers, i.e., all data moves directly from Storage I/O device Local Memory to Network I/O Device Local Memory.

Having an "interface file cache" that is "under control of said file system," as recited in claim 1, has advantages in unifying organization and control of files of the "interface file cache" with other files organized and controlled by the file system. Wang does not teach or suggest such advantages, and the Final Rejection does not attempt to explain how a modification of Wang to arrive at these features recited in claim 1 could have been obvious. Instead, the Final Rejection chooses to ignore these and other distinctions between Wang and claim 1 by stating, on page 8:

> First, the Examiner notes that the term "interface file cache" does not have a standard meaning in the art. Applicant has further not explicitly defined the term in the specification. Therefore, any cache (i.e., temporary memory) performing the functions claimed is sufficient to anticipate an "interface file cache." As noted in the cited definition from techdictionary.com, a cache is "a temporary memory area for frequently-accessed or recently-accessed data."

Appellants appreciate the admission by the Examiner that the term "interface file cache" does not have a standard meaning in the art, suggesting the novelty of that feature. As noted in *Phillips v. AWH Corp.*, 415 F.3d 1303, , (Fed. Cir. 2005) (*en banc*), "the patent by its nature describes something novel. See *Autogiro*, 384 F.2d at 397 ("Often the invention is novel and words do not exist to describe it. The dictionary does not always keep abreast of the inventor. It cannot.")."

Appellants respectfully disagree, however, with the Examiner's assertion that "Applicant has further not explicitly defined the term in the specification. Therefore, any cache (i.e., temporary memory) performing the functions claimed is sufficient to anticipate an "interface file cache." As noted in *Phillips*:

> Assigning such a limited role to the specification, and in particular requiring that any definition of claim language in the specification be express, is inconsistent with our rulings that the specification is "the single best guide to the meaning of a disputed term."

Moreover, as noted above, Wang depicts and describes a "file cache," demonstrating an understanding by Wang of this term. The Examiner chooses to ignore this fact, probably because the "file cache" that is described and depicted in Wang teaches away from the claimed limitation. The Examiner instead states that "any cache (i.e., temporary memory) performing the functions claimed is sufficient to anticipate an 'interface file cache,'" further ignoring the fact that the element cited by the Examiner is not "under control of said file system," as noted above. The Examiner then selects a dictionary to define the word "cache," instead of the term "interface file cache." As further noted in *Phillips*:

> Properly viewed, the "ordinary meaning" of a claim term is its meaning to the ordinary artisan after reading the entire patent. Yet heavy reliance on the dictionary divorced from the intrinsic evidence risks transforming the meaning of the claim term to the artisan into the meaning

> of the term in the abstract, out of its particular context, which is the specification.

In addition, the dictionary definitions chosen by the Examiner are stated on page 13 of the Final Rejection to be from "techdictionary.com, visited 2/17/05," as opposed to definitions that existed at the time of the invention. This also is contrary to another mandate of *Phillips*, which states:

> We have made clear, moreover, that the ordinary and customary meaning of a claim term is the meaning that the term would have to a person of ordinary skill in the art in question at the time of the invention, i.e., as of the effective filing date of the patent application.

Therefore, the Examiner has erred not only in elevating his selected dictionary definitions above the intrinsic evidence, but in choosing to proffer dictionary definitions from the wrong time period.

Moreover, the Examiner contradicts his allegation that the "Peer I/O Manager" is "a file system," by later alleging that one of ordinary skill in the art would have substituted a UDP socket in place of the 'Peer I/O Manager's unique Socket ID' that is taught by Wang (see column 11, lines 19-21). Appellants respectfully assert that "a file system" does not have a "unique Socket ID" as the Examiner alleges, but instead can manage data for various applications.

In sum, appellants respectfully assert that the Final Rejection's strained and convolution construction of the "interface file cache" recited in claim 1 ignores the words "interface" and "file," ignores the intrinsic evidence from appellants' specification, ignores the teachings of Wang and ignores the recital in claim 1 that the "interface file cache" is "under control of said file system." After ignoring such relevant evidence, the Final Rejection chooses to define a selected word from the claim with a dictionary definition that is inappropriate in both time and context in order to allege that Wang meets the recited limitations. For at least these reasons, the Final Rejection fails to state a *prima facie* case of obviousness of claim 1.

The Final Rejection also fails to show that Wang teaches or suggest the recital from claim 1 "wherein said host computer is configured to designate a User Datagram Protocol socket that is accessible by said interface device, and said interface device is configured to communicate said data between the network and the file cache according to

said User Datagram Protocol socket." The Final Rejection admits that "Wang does not explicitly teach that the socket may be a User Datagram Protocol (UDP) socket," and then cites a passage and figure from Wang ("col. 4, line 38 – col. 5, line 5; Fig. 3") that does not teach or suggest a "socket." For this reason also the Final Rejection fails to state a *prima facie* case of anticipation or obviousness of claim 1.

To attempt to remedy this additional failing, the Examiner introduces a definition for the word "socket" that he has selected from a dictionary. As with other definitions chosen by the Examiner, his proposed dictionary definition of the word "socket" is dated well after the filing date of the application, and thus is not evidence of ordinary meaning in the art at the time of the invention. Indeed, even the Examiner's specially selected and untimely dictionary definition lists several definitions for the term "socket," from which the Examiner has chosen his favorite. His selected definition also ignores the specification, which states, on page 31, lines 19-27:

> Unlike TCP, UDP does not offer a dependable connection. Instead, UDP packets are sent on a best efforts basis, and packets that are missing or damaged are not resent, unless a layer above UDP provides for such services. UDP provides a way for applications to send data via IP without having to establish a connection. A socket, however, may initially be designated in response to a request by UDP or another protocol, such as network file system (NFS), TCP, RTCP, SIP or MGCP, the socket allocating a port on the receiving device that is able to accept a message sent by UDP. A socket is an application programming interface (API) used by UDP that denotes the source and destination IP addresses and the source and destination UDP ports.

Appellants also respectfully disagree with the Examiner's assertion that "a socket is ... required for data transfer on a network." Appellants respectfully assert that many data transfers, such as ICMP messages, do not need port numbers and therefore do not require the Examiner's definition of a socket. Along those lines, appellants respectfully disagree with the Examiner's assertion that "As understood by one of ordinary skill in the art at the time of the invention, when UDP is used, UDP sockets are inherently required in order to receive the information packets over the network (see cited definition from techdictionary.com)." Initially note that, as mentioned above, the definitions from "techdictionary.com" that were cited by the Examiner were not from the time of the invention. Further note that the Examiner's cited definition of "UDP" from

"techdictionary.com" does not mention the term "socket," and the Examiner's cited definition of "socket" from "techdictionary.com" does not mention the term "UDP," so that it is curious that the Examiner would base his allegation that "UDP sockets are inherently required" on definitions that say no such thing.

Thus, the Final Rejection's interpretation of the "wherein" clause recited in claim 1 also avoids the intrinsic evidence from appellants' specification and the teachings of Wang to define selected words from the claim with dictionary definitions that are inappropriate in both time and context in order to allege that Wang meets the recited limitations. For at least these additional reasons, the Final Rejection fails to state a *prima facie* case of anticipation or obviousness of claim 1.

2) One of ordinary skill in the art would not have modified Wang as proposed by the Final Rejection.

Further evidence of the nonobviousness of claim 1 over the cited references is found in the Final Rejection's proposed modification of Wang that attempts to read on claim 1. Initially note that in proposing his untimely dictionary definition of the term "socket," the Examiner chooses to ignore the teachings of Wang, which states in column 11, lines 19-20, that "The Peer I/O Manager, like NCP, is a protocol layer that accepts data packets addressed to its own unique Socket ID in a file server." The Examiner probably avoids this teaching of Wang because, in an attempt to argue that claim 1 is obvious, the Final Rejection essentially proposes that one of ordinary skill in the art would have substituted a RTP protocol layer in place of the "Peer I/O Manager ... protocol layer" that is taught by Wang, and would have substituted a RTP socket ID in place of the "unique Socket ID" of Wang's "Peer I/O Manager." It is not apparent, however, and would not have been clear to one of ordinary skill in the art, how to substitute RTP for the "Peer I/O Manager ... protocol layer" of Wang.

Assuming arguendo that such a substitution could have been made, it would have likely undermined if not destroyed the "Peer I/O Manager's" alleged ability to "bypass the planar board memory," because RTP does not provide such an ability. Because this ability of the "Peer I/O Manager" is crucial to Wang, one of ordinary skill in the art would not have modified Wang as proposed by the Final Rejection. Moreover, had the

"Peer I/O Manager" been somehow replaced by RTP, it is not clear that Wang would have been able to function, as conflicting signals would likely have incapacitated the computer. Stated differently, the motivation proposed by the Examiner to make the modification proposed by the Examiner would have been overwhelmed by the disincentive apparent to one of ordinary skill in the art to destroy Wang by making the Examiner's modification.

Appellants also respectfully disagree with the Examiner's assertion on page 9 of the Final Rejection that it is "well known" to substitute one protocol for another. If this assertion is based the personal knowledge of the Examiner, appellants respectfully request that the Examiner provide an affidavit to support this assertion. While it may be easy to replace the name of one protocol with another on paper, one of ordinary skill in the art would recognize that such a substitution in practice may be difficult if not impossible, and may not function, such as the Examiner's proposed substitution of the "Peer I/O Manager ... protocol layer" with RTP.

For at least these reasons, one of ordinary skill in the art would not have been motivated to make the substitution proposed in the Final Rejection, and had such a substitution been made the resulting combination would have had substantial and nonobvious differences from the invention recited in claim 1.

3) The Final Rejection does not state a *prima facie* case of obviousness of claim 1.

In sum, several limitations of claim 1 are not taught or suggested in Wang, and the Final Rejection includes no explanation of why one of ordinary skill in the art would have modified Wang to provide those limitations. For example, an "interface file cache... under control of said file system" is not taught or suggested by Wang. In addition, one of ordinary skill in the art would not have made the modification proposed by the Final Rejection, and if such modification had been made, the apparatus defined by claim 1 would have further nonobvious differences from that proposedly modified device. For at least these reasons, the Final Rejection has failed to state a *prima facie* case of obviousness of claim 1.

B. Claim 3

Regarding claim 3, the Final Rejection states:

> As for claim 3, Wang does not explicitly disclose the use of Realtime Transport Protocol (RTP) headers. Schulzrinne teaches creating RTP headers and prepending the header to the data for transmission over the network (Sections 5, 5.1, 10). It would have been obvious to one of ordinary skill in the art at the time of the invention to modify Wang by creating RTP headers and prepending the header to data in order to maintain the real-time characteristics of data such as audio and video, as taught by Schulzrinne above.

Claim 3, however, recites:

> The apparatus of claim 1, wherein said host computer is configured to create a Realtime Transport Protocol header that is accessible by said interface device, and said interface device is configured to prepend said Realtime Transport Protocol header to said data.

Neither Schulzrinne nor Wang teaches or suggests that "said host computer is configured to create a Realtime Transport Protocol header" and that "said interface device is configured to prepend said Realtime Transport Protocol header to said data." On page 11 of the Final Rejection, the Examiner contends that these features are "inherent to the use of RTP and UDP and, in fact, *explicitly taught*" by Schulzrinne. Appellants respectfully disagree. *Assuming arguendo* that one of ordinary skill in the art would have modified Wang to use RTP as proposed by the Final Rejection with regard to claim 1, appellants respectfully assert that it is not "inherent to the use of RTP" described by Schulzrinne to use "an interface device (that) is configured to prepend said Realtime Transport Protocol header to said data.." As stated in *Continental Can Co. v. Monsanto Co.*, 948 F.2d 1264, 1269 (Fed. Cir. 1991):

> Inherency, however, may not be established by probabilities or possibilities. The mere fact that a certain thing may result from a given set of circumstances is not sufficient. [Citations omitted.]

Appellants respectfully assert that, *assuming arguendo* that one of ordinary skill in the art would have modified Wang to use RTP as proposed by the Final Rejection with regard to claim 1, one of such skill may have instead have used a host computer to prepend a RTP header, or used an interface device to create a RTP header, in contrast to claim 3. The Final Rejection provides no reason for the rejection of claim 3, other than

that the limitations of claim 3 are "inherent" and "explicitly taught," both of which are untrue. For these additional reasons, claim 3 is nonobvious over the references cited in the Final Rejection.

C. Claim 4

Regarding claims 4, 5 and 6, the Final Rejection states:

> As for claims 4, 5 and 6, Wang does not explicitly disclose the use of UDP headers. Stevens teaches that UDP, by definition, prepends data with UDP headers, wherein the data is further divided into plural fragments which are concatenated corresponding to the UDP header (Sections 11.2 and 11.5). It would have been obvious to one of ordinary skill in the art at the time of the invention to modify Wang by using UDP headers and dividing the data into plural fragments, in order to efficiently transfer data over a network, as taught by Stevens above.

Claim 4, however, recites:

> The apparatus of claim 1, wherein said data is stored with an associated User Datagram Protocol header, and said interface device includes a mechanism configured to process said User Datagram Protocol header.

Appellants respectfully assert that neither Stevens nor Wang teaches or suggests that "said interface device includes a mechanism configured to process said User Datagram Protocol header." For this additional reason, claim 4 is nonobvious over the references cited in the Final Rejection.

Responsive to appellants' earlier argument on this point, the Final Rejection on page 11 argues that "protocol stacks (see definition from techdictionary.com) located in the respective network interfaces process the headers in order to packetize (sending side) and concatenate (receiving side) the data packets." Wang, however, never mentions a "protocol stack" located in its "network I/O device." Appellants thus object to the Examiner's selected dictionary definition of "protocol stack" as not only untimely and inappropriate but irrelevant. Instead, Wang depicts "NETWORK PROTOCOLS 46" in FIG. 5. According to Wang (see, e.g., column 7, line 19): "The network protocols 46 is an efficient network protocol." According to Stevens, however, UDP is not a "network protocol," but is instead a "transport protocol." Appellants therefore respectfully

disagree with the Examiner's allegation that "the interface device of Wang would *necessarily* comprise just such a protocol stack for performing these functions."

Appellants respectfully assert that it is not trivial, and would not have been obvious to one of ordinary skill in the art, to modify Wang by adding transport layer protocol processing to its "network I/O device." Appellants respectfully assert that, *assuming arguendo* that one of ordinary skill in the art would have modified Wang to use UDP as proposed by the Final Rejection with regard to claim 1, one of such skill may have used "planar board 20 having main CPU(s) 24 and main dynamic memory 22" of Wang to process a UDP header, instead of a "network I/O device," in contrast to claim 4. The Final Rejection provides no reason for the rejection of claim 4, other than that the limitations of claim 4 are "inherent" and "explicitly taught," both of which are untrue. For these additional reasons, claim 4 is nonobvious over the references cited in the Final Rejection.

D. Claim 5

The Final Rejection of claims 4, 5 and 6 is shown above. Claim 5, however, recites:

> The apparatus of claim 1, wherein said data is prepended with a User Datagram Protocol header by said interface device to create a User Datagram Protocol datagram, and said interface device includes a mechanism configured to divide said datagram into plural fragments.

It is clear from Wang that the "network I/O device" only processes the network layer protocol, not transport layer protocols such as UDP, as discussed above. Similarly, column 8, lines 6-9 of Wang state:

> The Network Protocol 46 software component on the network I/O device 40 is responsible for creating network layer data packets using the raw file data and sends the data to Network Attached Clients 12.

Appellants respectfully assert that, *assuming arguendo* that one of ordinary skill in the art would have modified Wang to use UDP as proposed by the Final Rejection with regard to claim 1, one of such skill may have used "planar board 20 having main CPU(s) 24 and main dynamic memory 22" of Wang to prepend a UDP header, instead of a "network I/O device," in contrast to claim 5. The Final Rejection provides no reason for

the rejection of claim 5, other than that the limitations of claim 5 are "inherent" and "explicitly taught," both of which are untrue. For these additional reasons, claim 5 is nonobvious over the references cited in the Final Rejection.

E. Claim 6

The Final Rejection of claims 4, 5 and 6 is shown above. Claim 5, however, recites:

> The apparatus of claim 1, wherein said data is disposed in plural fragments, and said interface device includes a mechanism configured to concatenate said fragments corresponding to a User Datagram Protocol header.

Appellants respectfully assert that, *assuming arguendo* that one of ordinary skill in the art would have modified Wang to use UDP as proposed by the Final Rejection with regard to claim 1, the "network I/O device," would not have included "a mechanism configured to concatenate said fragments corresponding to a User Datagram Protocol header," in contrast to claim 6. Wang does not specifically teach how to receive packets from a network, instead stating in column 9, lines 10-17, that:

> The initial implementation of Peer I/O Manager 26 is designed to expedite file read operations, because it is believed that read operations are perceived as more time critical than write operations. This is true for applications such as video playback. However, the basic preferred implementation is fully capable of performing file write operations through Peer I/O.

Appellants respectfully assert, however, that write operations are much more complicated, because the data can be out of order or erroneous when received from the network, unlike read operation data that are under complete control of the host computer. That is, parsing, sorting, validation, reassembly of out of order segments, etc. are not issues for sending but are issues for receiving data. For this reason, *assuming arguendo* that Wang would have been modified as proposed in the Final Rejection, the resulting combination would not teach that "said interface device includes a mechanism configured to concatenate said fragments corresponding to a User Datagram Protocol header," in contrast to claim 6.

Even for the simpler case of sending data, however, Wang makes clear that only network layer protocols, not transport layer protocols such as UDP, are handled by its "network I/O device." For example, according to column 12, lines 57-63 of Wang:

> In response to file read requests, the Peer I/O Manager transmits file data to the client redirector extension-based clients via raw IPX data packets. This means that no higher layer protocol information is contained in the IPX--Data field. The client redirector extension relies upon the SocketID in the IPX Header to properly assemble and process the received data.

For this additional reason, *assuming arguendo* that Wang would have been modified as proposed in the Final Rejection of claim 1, the resulting combination would not teach that "said interface device includes a mechanism configured to concatenate said fragments corresponding to a User Datagram Protocol header," in contrast to claim 6. The Final Rejection provides no explanation for the rejection of claim 6, other than that the limitations of claim 6 are "inherent" and "explicitly taught," both of which are untrue. For these additional reasons, claim 6 is nonobvious over the references cited in the Final Rejection.

F. Claim 7

Regarding claim 7, the Final Rejection states:

> As for claim 7, Wang teaches the apparatus of claim 1, wherein said data does not enter said host computer (col. 3, lines 31-52; Fig. 3).

Appellants respectfully assert that, *assuming arguendo* that Wang would have been modified as proposed in the Final Rejection, the resulting combination would not operate such that "said data does not enter said host computer," in contrast to claim 7. This is because, as discussed above, Wang teaches that "The client redirector extension relies upon the SocketID in the IPX Header to properly assemble and process the received data," yet the Final Rejection proposes replacing the "unique Socket ID" of Wang's "Peer I/O Manager" with UDP/RTP. Appellants respectfully assert that with the "unique Socket ID" of Wang's "Peer I/O Manager" replaced, Wang would not operate such that "said data does not enter said host computer," in contrast to claim 7.

G. Claim 11

Regarding claim 11, the Final Rejection states:

> As for claim 11, Wang discloses an apparatus for transferring information between a network and a peripheral device, the apparatus comprising:
>
> a host computer having a processor(CPU 24, Fig.2) connected to a host memory (memory 32, Fig.2) by a host memory bus (connection illustrated in Fig.2), said host memory containing an application operable by the processor to designate a *socket* (Note, a socket is merely an endpoint of a connection, which is inherently required for communications on a network. See cited definition from techdictionary.com.; col. 4, line 38 – col. 5, line 5; Fig. 3), and
>
> an interface device (network I/O device 40, Fig. 2) connected to said host computer and coupled between the network and the peripheral device, said interface device including an interface memory adapted to store data corresponding to *said socket* and a mechanism configured to associate said data with *a header* corresponding to *said socket* such that said data is communicated between the network and the peripheral device without encountering said host computer (col. 4, line 38 – col. 5, line 5; Fig. 3).
>
> Although obvious to one of ordinary skill in the art at the time of the invention, Wang does not explicitly teach that the socket may be a Uniform Datagram Protocol (UDP) socket. The Examiner notes that UDP is a well-known protocol for data transfer on networks, as shown by Stevens, chapter 11, for example. Schulzrinne further teaches the use of UDP as an underlying protocol for RTP in order to maintain the real-time characteristics of data such as audio and video (section 1, Introduction). As understood by one of ordinary skill in the art at the time of the invention, when UDP is used, UDP sockets are inherently required in order to designate the endpoints of the connection (See cited definition from techdictionary.com). It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the teachings of Wang by using UDP sockets in order to maintain the real-time characteristics of data such as audio and video, as taught by Schulzrinne above.

Applicants respectfully disagree with the Final Rejection assertion that it "would have been obvious to one of ordinary skill in the art at the time of the invention to modify the teachings of Wang by using UDP sockets in order to maintain the real-time characteristics of data such as audio and video, as taught by Schulzrinne above." As best as can be understood, the Final Rejection is essentially proposing that one of ordinary skill in the art would have substituted a RTP protocol layer in place of the "Peer I/O

Manager...protocol layer" of Wang, and would have substituted a RTP socket ID in place of the "unique Socket ID" of the "Peer I/O Manager" that is taught by Wang. Appellants respectfully assert that one of such skill would not have made such a modification because it would have at best been unclear how such a substitution could have been made, and because it is likely that the resulting device probably would not function to transfer data. Moreover, the resulting device would not have had the functions provided by the "Peer I/O Manager" that are described as advantageous by Wang. For example, replacing the "Peer I/O Manager software component 26," as proposed by the Final Rejection, would remove "the program control necessary to co-ordinate and initiate Peer I/O data transfers from the Storage I/O Device 50 to the Network I/O Device 40." (see column 6, line 66 – column 7, line 2). For at least these reasons, one of ordinary skill in the art would not have been motivated to make the modification proposed by the Final Rejection.

In addition, had the modification of Wang proposed by the Final Rejection been made, the functions provided by the "Peer I/O Manager" would have been replaced, and Wang would not, contrary to the Final Rejection assertions, have data that "is communicated between the network and the peripheral device without encountering said host computer." Appellants also respectfully disagree with the Examiner's assertion on page 9 of the Final Rejection that it is "well known" to substitute one protocol for another. If this assertion is based the personal knowledge of the Examiner, appellants respectfully request that the Examiner provide an affidavit to support this assertion. While it may be easy to replace the name of one protocol with another on paper, one of ordinary skill in the art would recognize that such a substitution in practice may be difficult if not impossible, and may not function, such as the Examiner's proposed substitution of the "Peer I/O Manager ... protocol layer" with RTP.

In short, one of ordinary skill in the art at the time of the invention would not have been motivated to make the modification of Wang that is proposed in the Final Rejection, and even if such modification had been attempted, the apparatus defined by claim 11 would have substantial and nonobvious differences from the modification of Wang with Stevens and Schulzrinne that is proposed in the Final Rejection.

H. Claim 12

Regarding claim 12, the Final Rejection states:

> As for claim 12, Wang discloses the apparatus of claim 11, wherein said host computer contains a file system (file system 27, Fig. 3) and said interface memory includes a file cache (local memory, 44, Fig. 3) adapted to store said data, wherein said file system manages storage of said data in said file cache (col. 4, line 38 – col. 5, line 5; Fig. 3).

Initially note that the "file system 27, Fig. 3" mentioned by the Examiner is separate from the PEER I/O MANAGER 26 in FIG. 3. Further note that the "local memory, 44, Fig. 3" is not described or depicted as a "file cache" in Wang, in contrast to "FILE CACHE 4a" shown in FIG. 5 of that reference. The fact that Wang describes and depicts a "file system" that is separate from and different than the "Peer I/O Manager" alleged by the Final Rejection to be a "file system," and that Wang describes and depicts a "file cache" that is separate from and different than the "local memory, 44, Fig. 3" alleged by the Final Rejection to be a "file cache," argues that the Examiner's interpretation of those terms is misplaced.

Assuming arguendo that Wang teaches a "file cache (local memory, 44, Fig. 3)," appellants further respectfully disagree with the Final Rejection assertion that "said (file system 27, Fig. 3) manages storage of said data in said file cache (col. 4, line 38 – col. 5, line 5; Fig. 3)." The passage cited by the Final Rejection, as well as the rest of Wang (see, e.g., column 6, line 66 – column 7, line 4), makes clear that, as stated in column 4, lines 1-5 of Wang:

> The Peer I/O Manager implementation provides the program control necessary for coordinating and initiating peer I/O transfers directly from the storage I/O device to the network I/O device.

As previously discussed, the Peer I/O Manager is defined by Wang in column 11, lines 19-23:

> The Peer I/O Manager, like NCP, is a protocol layer that accepts data packets addressed to its own unique Socket ID in a file server. The client redirector extension is a client-based software protocol layer that addresses file requests to the Peer I/O Manager socket ID in the file server.

Such protocol layers and Socket IDs are discussed by Wang in column 11, lines 8-18:

> The IPX layer uses the Destination Socket in the IPX Header field of each incoming packet as the postal address that associates incoming data packets with specific higher layer protocols. Protocols register with IPX to receive packets addressed to specific Socket IDs.
>
> FIG. 7, illustrates the NetWare™ Core Protocol (NCP) as a software layer above IPX. IPX delivers data packets addressed to Socket number 451 to NCP. Traditionally, NetWare™ client-based redirectors send file requests to the NCP socket ID in the file server. (emphasis added)

In short, the "Peer I/O Manager" of Wang is a "protocol layer that accepts data packets addressed to its own unique Socket ID," and is not a "file system" that "manages storage of said data in said file cache," in contrast to claim 12.

Moreover, *assuming arguendo* that one of ordinary skill would have made the modification of Wang proposed by the Final Rejection in order to attempt to read on claim 11, claim 12 would be even further removed from obviousness because Wang's "Peer I/O Manager... protocol layer" and "its own unique Socket ID" would be replaced, as noted above, by the Examiner's proposed protocol layers. Because Wang as proposedly modified does not teach or suggest an "interface memory (that) includes a file cache ... wherein said file system manages storage of said data in said file cache" as recited in claim 12, claim 12 is not obvious over those references.

I. Claim 13

Regarding claims 13 and 14, the Final Rejection states:

> As for claims 13 and 14, Wang does not explicitly disclose the use of UDP packets and headers. Stevens teaches that UDP, by definition, includes UDP packets and headers, wherein said data travels over the network in plural fragments (packets) corresponding to the header. The interface device is further required to process the UDP headers and concatenate the data. See Stevens, Chapter 11, and particularly sections 11.2 and 11.5. It would have been obvious to one of ordinary skill in the art at the time of the invention to modify Wang by using UDP packets and headers, wherein the interface device processes the headers and concatenates the data, because these are well-known and necessary steps in order to efficiently transfer data over a network, as taught by Stevens above.

Claim 13, however, recites:

> The apparatus of claim 11, wherein said data travels over the network in at least one packet containing a User Datagram Protocol header, and said interface device includes circuitry configured to process said User Datagram Protocol header.

There is no teaching or suggestion in Wang or Stevens that "said interface device includes circuitry configured to process said User Datagram Protocol header," as recited in claim 13. Wang does not even mention the words "circuit" or "circuitry," and such circuitry is not inherent in Wang. The Final Rejection also does not assert that Wang or any other cited reference teaches such circuitry, and for at least these reasons fails to state a *prima facie* case of obviousness of claim 13.

I. Claim 14

Similarly, claim 14 recites:

> The apparatus of claim 11, wherein said data travels over the network in plural fragments corresponding to a User Datagram Protocol header, and said interface device is configured to concatenate said data with said User Datagram Protocol header.

There is no teaching or suggestion in Wang, Schulzrinne or Stevens that "said interface device is configured to concatenate said data with said User Datagram Protocol header," as recited in claim 14. Applicants respectfully disagree with the Final Rejection assertion that: "The interface device is further required to process the UDP headers and concatenate the data. See Stevens, Chapter 11, and particularly sections 11.2 and 11.5." Applicants respectfully request the Examiner to point out where in Chapter 11 Stevens teaches that an "interface device" is "required to process the UDP headers and concatenate the data." The Final Rejection on page 11 avoids this question by stating that Wang teaches "an 'interface device' (network I/O device, Fig.3)" but Wang also teaches that the "network I/O device" does not process transport layer headers such as UDP. The Examiner then states that "the use of an interface device is *inherent*..." and that it "is not possible to send and receive data packets over a network unless there are at least two interfaces devices for performing the respective sending and receiving." Appellants respectfully assert that it is well known that a host computer to send or receive packets over a network without an "interface device," again disabusing the Examiner's definition

of inherency. For these additional reasons, the Final Rejection fails to stat a prima facie case of obviousness.

J. Claim 15

Regarding claim 15, the Final Rejection states:

> As for claim 15, Wang does not explicitly disclose the use of Realtime Transport Protocol (RTP). Schulzrinne teaches the use of RTP and RTP headers in order to transfer data over a network while maintaining the real-time characteristics (Section 1, Introduction; Section 5, 5.1, RTP fixed header fields). It would have been obvious to one of ordinary skill in the art at the time of the invention to modify Wang by using RTP and RTP headers in order to transfer data over a network while maintaining the real-time characteristics, as taught by Schulzrinne above.

Claim 15, however, recites:

> The apparatus of claim 11, wherein said host computer is configured to create a Realtime Transport Protocol header that is accessible by said interface device, and said interface device is configured to prepend said Realtime Transport Protocol header to said data.

Appellants respectfully assert that none of the cited references teaches or suggests that "said host computer is configured to create a Realtime Transport Protocol header" and that "said interface device is configured to prepend said Realtime Transport Protocol header to said data." On page 11 of the Final Rejection, the Examiner contends that these features are "inherent to the use of RTP and UDP and, in fact, *explicitly taught*" by Schulzrinne. Appellants respectfully disagree. *Assuming arguendo* that one of ordinary skill in the art would have modified Wang to use RTP as proposed by the Final Rejection with regard to claim 11, appellants respectfully assert that it is not at all "inherent to the use of RTP" described by Schulzrinne to use "an interface device (that) is configured to prepend said Realtime Transport Protocol header to said data.."

Appellants respectfully assert that, *assuming arguendo* that one of ordinary skill in the art would have modified Wang to use RTP as proposed by the Final Rejection with regard to claim 11, one of such skill may have instead have used a host computer to prepend a RTP header, or used an interface device to create a RTP header, in contrast to claim 15. The Final Rejection provides no reason for the rejection of claim 15, other than that the limitations of claim 15 are "inherent" and "explicitly taught," both of which are

untrue. For these additional reasons, claim 15 is nonobvious over the references cited in the Final Rejection.

K. Claim 21

The Final Rejection does not distinguish claim 21 from claim 1, despite the "means-plus-function" clause in claim 21. In particular, the Examiner disregards the structure disclosed in appellants' specification corresponding to the "means-plus-function" clause in claim 21 in rendering his opinion that claim 21 is not patentable. Because the Examiner does not attempt to show that the structure corresponding to the "means for" clause in claim 21 is obvious over the cited references, the Final Rejection has failed to present a *prima facie* case of obviousness of claim 21.

In addition, appellants incorporate by reference the arguments made above with regard to claim 1, to further explain why claim 21 is not obvious over the cited references.

II. Regarding Grounds of Rejection (2), the Final Rejection states:

> **Claims 2 and 22** are rejected under 35 U.S.C. 103(a) as being unpatentable over Wang in view of Stevens and Schulzrinne and further in view of Applicant's admitted prior art (pg. 31, line 28 – pg. 32, line 8) (hereinafter AAPA).
>
> As for claims 2 and 22, Wang explicitly teaches the use of application layer headers, which inherently includes prepending these headers to the data (col. 11, lines 14-36), because otherwise the data packets could not be transferred. It is not clear from Wang whether or not these application layer headers are created by the host computer or the interface device. Thus, Wang, Stevens and Schulzrinne do not specifically disclose a host computer that is configured to create an application layer header that is accessible by said interface device. However, AAPA (pg. 31, line 28 – pg. 32, line 8) teaches that it is conventionally known to generate an application packet header at a host computer. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the teachings of Wang and Schulzrinne by configuring the host computer to create an application layer header that is accessible by said interface device in order to transmit application data over the network. Moreover, it would have been obvious to generate the application layer headers at the host computer of Wang, because the application resides on the host computer and this would simplify data processing.

A. Claim 2

Regarding claim 2, the Final Rejection states:

> As for claims 2 and 22, Wang explicitly teaches the use of application layer headers, which inherently includes prepending these headers to the data (col.11, lines 14-36), because otherwise the data packets could not be transferred. It is not clear from Wang whether or not these application layer headers are created by the host computer or the interface device. Thus, Wang, Stevens and Schulzrinne do not specifically disclose a host computer that is configured to create the application layer header that is accessible by the interface device. However, AAPA (pg. 31, line 28 – pg. 32, line 8) teaches that it is conventionally known to generate an application packet header at a host computer. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the teachings of Wang and Schulzrinne by configuring the host computer to create an application layer header that is accessible by said interface device in order to transmit application data over the network. Moreover, it would have been obvious to generate the application headers at the host computer of Wang, because the application resides on the host computer and this would simplify data processing.

Claim 2, however, recites:

> The apparatus of claim 1, wherein said host computer is configured to create an application layer header that is accessible by said interface device, and said interface device is configured to prepend said application layer header to said data.

Regarding claim 2, the Final Rejection fails to show the existence of any incentive in the cited references for why "it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the teachings of Wang and Schulzrinne by configuring the host computer to create an application layer header that is accessible by said interface device in order to transmit application data over the network." Instead, the Final Rejection admits that "Wang, Stevens and Schulzrinne do not specifically disclose a host computer that is configured to create an application layer header that is accessible by said interface device." The passage from appellants' specification that is cited by the Final Rejection does not state that the prior art includes such a motivation. For at least these reasons claim 2 is not obvious over the cited references.

In addition, the Final Rejection does not even attempt to show that other limitations of claim 2, for example that "said interface device is configured to prepend

said application layer header to said data," are taught or suggested by the cited references. In addition, no motivation to modify the cited references to include such limitations is provided by the Final Rejection.

Appellants further respectfully disagree with the Examiner's statement that "Wang explicitly teaches the use of application layer headers, which inherently includes prepending these headers to the data (col. 11, lines 14-36), because otherwise the data packets could not be transferred." Appellants respectfully assert that data packets can be transferred without application layer headers.

Moreover, appellants respectfully disagree with the Examiner's statement that "AAPA (pg. 31, line 28 – pg. 32, line 8) teaches that it is conventionally known to generate an application packet header at a host computer." Indeed, appellants are uncertain what is even meant by the Final Rejection's reference to "an application packet header."

For at least these various reasons, the Final Rejection fails to state a *prima facie* case of obviousness of claim 2.

B. Claim 22

Claim 22 recites:

> The apparatus of claim 1, wherein said host computer is configured to create an application layer header that is accessible by said interface device, and said interface device is configured to prepend said application layer header to said data.

The Final Rejection does not distinguish claim 22 from claim 2, despite the "means-plus-function" clause in claim 21, from which claim 22 depends. Because the Examiner does not attempt to show that the structure corresponding to the "means for" clause in claim 21 is obvious over the cited references as implicated in claim 22, the Final Rejection has failed to present a *prima facie* case of obviousness of claim 22.

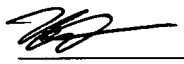
In addition, appellants incorporate by reference the arguments made above with regard to claim 2, to further explain why claim 22 is not obvious over the cited references.

Conclusion

As detailed above, the Final Rejection fails to state a *prima facie* case of obviousness for any of the pending claims. Appellants respectfully assert that all the pending claims are allowable and request reversal of the Examiner's rejections.

This brief is being submitted along with a check in the amount of $500.00 to pay the Appeal Brief Fee.

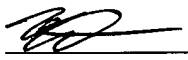
Respectfully submitted,



Mark Lauer
Reg. No. 36,578
6601 Koll Center Parkway
Suite 245
Pleasanton, CA 94566
Tel: (925) 484-9295
Fax: (925) 484-9291

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to Mail Stop Appeal Brief, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on August 24, 2005.

Date: 8-24-05 

Mark Lauer

APPENDIX A – CLAIMS ON APPEAL

1. An apparatus for transferring information between a network and a storage device, the apparatus comprising:

a host computer having a CPU operating a file system and a host memory connected to said CPU by a host bus, and

an interface device coupled to said host computer, to the network and to the storage device, said interface device including an interface memory containing an interface file cache adapted to store data that is communicated between the network and the storage device under control of said file system,

wherein said host computer is configured to designate a User Datagram Protocol socket that is accessible by said interface device, and said interface device is configured to communicate said data between the network and the file cache according to said User Datagram Protocol socket.

2. The apparatus of claim 1, wherein said host computer is configured to create an application layer header that is accessible by said interface device, and said interface device is configured to prepend said application layer header to said data.

3. The apparatus of claim 1, wherein said host computer is configured to create a Realtime Transport Protocol header that is accessible by said interface device, and said interface device is configured to prepend said Realtime Transport Protocol header to said data.

4. The apparatus of claim 1, wherein said data is stored with an associated User Datagram Protocol header, and said interface device includes a mechanism configured to process said User Datagram Protocol header.

5. The apparatus of claim 1, wherein said data is prepended with a User Datagram Protocol header by said interface device to create a User Datagram Protocol datagram, and said interface device includes a mechanism configured to divide said datagram into plural fragments.

6. The apparatus of claim 1, wherein said data is disposed in plural fragments, and said interface device includes a mechanism configured to concatenate said fragments corresponding to a User Datagram Protocol header.

7. The apparatus of claim 1, wherein said data does not enter said host computer.

8. The apparatus of claim 1, wherein said data includes audio data.

9. The apparatus of claim 1, wherein said data includes video data.

10. The apparatus of claim 1, wherein said data is a part of a realtime communication.

11. An apparatus for transferring information between a network and a peripheral device, the apparatus comprising:

a host computer having a processor connected to a host memory by a host memory bus, said host memory containing an application operable by the processor to designate a User Datagram Protocol socket, and

an interface device connected to said host computer and coupled between the network and the peripheral device, said interface device including an interface memory adapted to store data corresponding to said User Datagram Protocol socket and a mechanism configured to associate said data with a User Datagram Protocol header corresponding to said User Datagram Protocol socket such that said data is communicated between the network and the peripheral device without encountering said host computer.

12. The apparatus of claim 11, wherein said host computer contains a file system and said interface memory includes a file cache adapted to store said data, wherein said file system manages storage of said data in said file cache.

13. The apparatus of claim 11, wherein said data travels over the network in at least one packet containing a User Datagram Protocol header, and said interface device includes circuitry configured to process said User Datagram Protocol header.

14. The apparatus of claim 11, wherein said data travels over the network in plural fragments corresponding to a User Datagram Protocol header, and said interface device is configured to concatenate said data with said User Datagram Protocol header.

15. The apparatus of claim 11, wherein said host computer is configured to create a Realtime Transport Protocol header that is accessible by said interface device, and said interface device is configured to prepend said Realtime Transport Protocol header to said data.

16. The apparatus of claim 11, wherein said data includes audio data.

17. The apparatus of claim 11, wherein said data includes video data.

18. The apparatus of claim 11, wherein said data is a part of a realtime communication over the network.

21. An apparatus for transferring information between a network and a storage device, the apparatus comprising:

a host computer having a CPU operating a file system and a host memory connected to said CPU by a host bus, and

an interface device coupled to said host computer, to the network and to the storage device, said interface device including an interface memory containing an interface file cache adapted to store data that is communicated between the network and the storage device under control of said file system,

wherein said host computer is configured to designate a User Datagram Protocol socket that is accessible by said interface device, and said interface device has means for communicating said data between the network and the file cache according to said User Datagram Protocol socket.

22. The apparatus of claim 21, wherein said host computer is configured to create an application layer header that is accessible by said interface device, and said interface device is configured to prepend said application layer header to said data.